ISSN: 2755-6190

Research Article

Open Access Journal of Artificial Intelligence and Technology

VisionVerse: Dynamic Video Question Answering Through Retrieval-Augmented Generation

Abhiram S Sajeev*, Adhya Sanil Joseph, Amal Madhav T, Surekha Mariam Varghese and Aby Abahai T

Department of Computer Science and Engineering Mar Athanasius College of Engineering, Kothamangalam, Kerala, India

***Corresponding author**
Abhiram S Sajeev, Department of Computer Science and Engineering Mar Athanasius College of Engineering, Kothamangalam, Kerala, India.

Received: December 22, 2025; **Accepted:** December 29, 2025; **Published:** January 06, 2026

ABSTRACT

In the digital age, video content has become a distinguished form for information sharing, entertainment, and education. However, navigating and comprehending lengthy video content can be time-consuming and challenging for users. The project introduces an innovative solution that can sway state-of-the-art language models to transform extensive video content into concise text contents, making it more accessible and user- friendly. By leveraging Retrieval Augmented Generation(RAG), it accurately condenses videos into text form, ensuring that the core message and key details are retained. This process enhances the efficiency of content consumption by providing users with a quick, readable overview of the video's contents. Furthermore, introducing an interactive chatbot that enables users to engage with the video content. Users can ask questions, seek clarifications, or ferret in deeper into specific aspects of the video. The chatbot is powered by a Large Language Model, which enables meaningful and context-aware interactions. The idea not only facilitates better understanding but also encourages active participation and knowledge retention. Also, benefits of interactive chatbot and video summarization technologies together, offering users a dynamic and engaging means to access and interact with video content. The system employs advanced video-to-text summarization techniques to automatically extract the most relevant information from videos. This innovation has significant potential applications in education, research, and the digital content landscape, where the efficient dissemination of information is paramount.

Keywords: Video Summarization, RAG, Natural Language Processing, Chatbots, Information Retrieval, Human-Computer Interaction

Introduction

In the dynamic realm of the contemporary digital land- scape, video content has become a predominant and influential medium for the dissemination of information and entertainment alike. However, the omnipresence of video content is not without its challenges, especially concerning the efficient navigation and comprehension of extensive video materials. To address this issue, an innovative solution has emerged, influencing state-of-the-art language models to seamlessly trans form lengthy video content into concise and easily digestible text summaries. This revolutionary approach is meticulously designed to ensure the preservation of essential messages and key details, thereby offering users a more streamlined and user- friendly avenue for content consumption.

The transformative essence of this innovative solution lies in its adept utilization of advanced language models, showcasing unparalleled proficiency in natural language processing (NLP) tasks. At the core of this approach is the well thought out incorporation of modern language models, such as those powered by pioneering technologies like OpenAI's GPT-3. These models demonstrate exceptional capabilities in understanding and generating human-like text, enabling a sophisticated level of analysis and synthesis within the context of video summarization.

The efficiency in summarization achieved through the adept utilization of advanced language models directly addresses a critical challenge in the contemporary digital landscape time

Citation: Abhiram S Sajeev, Adhya Sanil Joseph, Amal Madhav T, Surekha Mariam Varghese, Aby Abahai T. VisionVerse: Dynamic Video Question Answering Through Retrieval-Augmented Generation. Open Access J Artif Intel Tech. 2026. 2(1): 1-9. DOI: doi.org/10.61440/OAJAIT.2026.v2.18

constraints. In a world where individuals are pounded with an abundance of content fighting for their attention, time has become a precious commodity. Lengthy videos, while rich in information, often pose a barrier to efficient knowledge acquisition, especially when users are constrained by busy schedules or competing priorities. By harnessing the capabilities of advanced language models, the system ensures that users can swiftly extract key insights from video content without the need for prolonged viewing sessions.

In essence, by providing a personalized and interactive platform for accessing and engaging with video content, this integrated solution stands poised to redefine the paradigm of information consumption in our increasingly digital-centric world. As the digital landscape continues to evolve, the trans formative potential of this innovation extends its reach across education, research, and the broader digital content landscape, promising a revolution in the way users navigate and derive insights from video content. remove unwanted content and make it to an introduction for publishing paper.

Related Works

Advancements in Natural Language Processing (NLP)

Recent years have witnessed significant advancements in natural language processing (NLP) technologies, driven by the development of sophisticated language models [1]. One notable example is OpenAI's GPT-3, a state-of-the-art language model capable of understanding and generating human-like text [2]. These advancements have paved the way for more sophisticated approaches to processing and summarizing textual content, including the summarization of video transcripts.

Video Summarization Techniques

Video summarization techniques have evolved to address the challenge of condensing lengthy video content into concise summaries while retaining key information [3]. Traditional methods include keyframe extraction, where representative frames are selected to summarize video content. However, with the rise of deep learning, newer approaches based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have gained prominence [3]. These deep learning-based methods leverage the temporal and spatial information present in video sequences to generate more informative summaries.

Interactive Platforms for Video Engagement

Interactive platforms for video engagement have emerged as a means to enhance user interaction and knowledge retention [4]. These platforms offer users various interactive features such as chatbots, quizzes, annotations, and discussion forums. By enabling users to engage with video content in a dynamic and participatory manner, these platforms foster deeper understanding and engagement [5]. Additionally, they provide valuable insights into user behavior and preferences, facilitating the design of more personalized and user centric experiences. These related works highlight the interdisciplinary nature of the research area, drawing from fields such as natural language processing, computer vision, and human-computer interaction. By integrating insights from these domains, researchers aim to develop innovative solutions that address the challenges associated with video content consumption in the digital age.

Proposed Model

This project basically consists of software modules.

Software Details

Python

Python's relevance for video-to-text summarization and chatbot creation is deeply rooted in its versatility and robust ecosystem. The language's widespread adoption in these domains is attributed to several key factors. Firstly, Python boasts powerful Natural Language Processing (NLP) libraries like NLTK (Natural Language Toolkit) and spacey, which empower developers to perform efficient text analysis a critical component for tasks such as summarization. These libraries provide tools for tokenization, part-of- speech tagging, and entity recognition, enabling the extraction of meaningful information from text, a prerequisite for creating accurate summaries.

Python's proficiency in data processing is also a significant advantage. For video-to-text summarization, Python can handle tasks such as speech-to-text conversion efficiently. Dedicated libraries like Genism provide tools for text summarization, allowing developers to distill key information from large textual datasets, including transcriptions of videos. In the realm of chatbot development, Python is well- supported by frameworks like Rasa, which offer a comprehensive set of tools for building context-aware and interactive chatbots. Python's readability and simplicity contribute to the ease of implementing and maintaining chatbot systems.

Streamlit

Streamlit, a dynamic Python library, empowers developers to effortlessly create interactive web applications for machine learning and data science projects. With its intuitive interface and simple syntax, Streamlit enables users to craft immersive data-driven experiences without the need for extensive web development knowledge. Leveraging Streamlit's versatile features, developers can seamlessly integrate data visualizations, interactive widgets, and machine learning models into their applications, facilitating streamlined analysis and enhanced user engagement. Whether it's proto typing ML algorithms, sharing insights with stakeholders, or deploying robust applications, Streamlit provides a seamless platform for transforming data into actionable intelligence. Its open-source nature fosters a vibrant community where developers can collaborate, innovate, and push the boundaries of what's possible in data-driven application development. From exploratory data analysis to production-ready solutions, Streamlit empowers developers to bring their ideas to life with unparalleled ease and efficiency, ushering in a new era of interactive data exploration and storytelling.

pytube

Pytube is a powerful Python library that facilitates easy access to YouTube videos and their metadata. With Pytube, developers can effortlessly retrieve information about videos, including titles, descriptions, durations, and more, making it an invaluable tool for tasks such as data collection, analysis, and automation. Moreover, Pytube sim- plifies the process of downloading YouTube videos directly to local storage, providing a convenient solution for various applications ranging from content creation to offline viewing. Its intuitive interface and extensive documentation make it accessible to developers of all skill levels,

enabling them to integrate YouTube functionality seamlessly into their Python projects. Whether extracting insights from video content or enhancing user experiences with multimedia capabilities, Py- tube empowers developers to harness the vast potential of YouTube's vast repository of content with ease and efficiency.

Lang Chain Framework

The Langchain framework plays a pivotal role in the fields of video-to-text summarization and chatbot development, offering a versatile set of features that make it highly significant in these domains. Its capabilities contribute to efficient natural language processing (NLP) and empower developers to tackle complex language-related tasks. Lang chain's strength lies in its comprehensive set of APIs, which streamline text analysis processes. These APIs enable developers to leverage a wide range of linguistic and semantic analysis tools, facilitating tasks like sentiment analysis, entity recognition, and language understanding. The framework's adaptability and extensibility contribute to its effectiveness in handling diverse video content and extracting relevant information accurately. In the domain of chatbot development, Langchain's modular architecture provides a robust foundation. Developers can create context-aware and interactive chatbots by harnessing the framework's NLP capabilities. Langchain's modules enable the chatbots to understand and respond to user inputs in a natural and meaningful way, enhancing the overall conversational experience. The modular design allows developers to customize and extend the functionality of chatbots based on specific project requirements.

Large Language Model (LLM)

Leveraging Large Language Models (LLMs), exemplified by advanced models like GPT-3, plays a pivotal role in the domains of video-to-text summarization and chatbot development [2]. The sophisticated natural language processing (NLP) capabilities inherent in LLMs enable precise extraction of key information from video content, offering efficiency and accuracy in the summarization process. Moreover, their vast linguistic understanding empowers chatbots to engage in contextually rich and coherent conversations, making them highly versatile for various language- related tasks. In video-to-text summarization, the advanced NLP capabilities of LLMs prove instrumental. These models can analyze spoken or written content within videos with a high degree of precision, facilitating the extraction of meaningful information. GPT-3, in particular, demonstrates proficiency in understanding context, identifying key themes, and summarizing content effectively. The utilization of LLMs streamlines the summarization workflow, ensuring that extracted information is both relevant and coherent.

Whisper

Whisper API is a robust platform that enables developers to integrate secure and scalable messaging functionality into their applications. With Whisper API, developers can facilitate real-time communication, ensuring privacy and confidentiality through end-to-end encryption and other security measures. The API provides a seamless interface for sending and receiving messages across various channels, including text, voice, and multimedia, empowering developers to create immersive and interactive user experiences. Additionally, Whisper API offers features such as message queuing, delivery status tracking, and user authentication, facilitating efficient and reliable communication workflows. Whether building messaging apps, collaboration platforms, or customer support systems, Whisper API equips developers with the tools they need to deliver seamless and secure communication solutions to their users.

Retrieval-Augmented Generation (RAG)

Retrieval- augmented generation refers to a methodology in natural language processing (NLP) where a model combines the capabilities of both retrieval-based and generation-based approaches [6]. This technique involves using a retrieval system to select relevant information from a large dataset or knowledge base and then leveraging a generation model to produce a response or generate text based on the retrieved information. By integrating retrieval and generation components, the model can effectively incorporate external knowledge and context into the generation process, resulting in more informative, coherent, and contextually relevant outputs. Retrieval-augmented generation has shown promise in various NLP tasks such as question answering, dialogue generation, and content summarization, where access to external knowledge sources enhances the quality and relevance of generated content. This approach represents a significant advancement in NLP, enabling models to leverage external information effectively and produce more accurate and contextually appropriate responses.

FAISS Vector DB

FAISS (Facebook AI Similarity Search) vector database is a powerful tool revolutionizing the field of large-scale similarity search and retrieval. Developed by Facebook AI Research, FAISS offers efficient indexing and search capabilities for high-dimensional vectors, making it ideal for applications such as image retrieval, natural language processing, and recommendation systems. By utilizing state- of-the-art algorithms and data structures, FAISS enables fast and scalable similarity search operations, allowing users to efficiently find nearest neighbors or retrieve vectors that are most similar to a given query. Its versatile design supports various indexing methods, including hierarchical clustering, product quantization, and inverted file indexing, providing flexibility and performance optimization for diverse use cases. FAISS's ability to handle massive datasets with millions or even billions of vectors while maintaining low latency and memory footprint makes it a go-to solution for organizations and researchers tackling complex similarity search challenges. With its open- source availability and active community support, FAISS continues to drive innovation in vector database technology, empowering developers to build cutting-edge applications that demand fast and accurate similarity retrieval.

HNSW Algorithm

The HNSW algorithm is a novel approach for approximate K-nearest neighbor (KNN) search, which is a fundamental problem in many areas, including machine learning, information retrieval, and computational biology [7]. KNN search aims to find the K closest data points to a given query point in a high-dimensional space, a computationally expensive task for large datasets. Traditional methods for KNN search often involve building complex tree-like data structures, such as KD-trees or ball trees, which can be inefficient for high-dimensional data due to the "curse of dimensionality." The HNSW algorithm takes a

different approach, utilizing a graph-based structure without the need for additional search structures typically employed in proximity graph techniques. The key idea behind HNSW is to incrementally build a multi-layer structure consisting of hierarchical sets of proximity graphs (layers) for nested subsets of the stored data points. Each data point is assigned to a maximum layer randomly, with an exponentially decaying probability distribution for higher layers.

This hierarchical structure allows for efficient navigation and search, similar to the previously studied Navigable Small World (NSW) graphs, but with an added benefit of separating links by their characteristic distance scales. The search process in HNSW starts from the top layer and utilizes the scale separation between layers, enabling logarithmic complexity scaling. Additionally, the algorithm employs a heuristic for selecting proximity graph neighbors, which significantly enhances performance at high recall rates and in the presence of highly clustered data. One of the notable advantages of HNSW is its strong performance compared to previous open-source state-of-the-art vector-only approaches for approximate KNN search. The algorithm's similarity to the skip list data structure allows for straightforward distributed implementation, making it suitable for large-scale datasets and parallel computing environments. In summary, the HNSW algorithm presents an innovative graph-based approach for approximate KNN search, offering efficient performance, scalability, and adaptability to high- dimensional data. Its hierarchical structure, scale separation, and heuristic neighbor selection contribute to its strong performance and potential for widespread adoption in various domains requiring KNN search capabilities.

Cascading Style Sheets

CSS, or Cascading Style Sheets, is an integral component of web development that plays a pivotal role in shaping the visual presentation and aesthetics of HTML elements within a webpage. Serving as the stylistic language of the web, CSS empowers developers to exercise precise control over layout, colors, fonts, and various other design aspects, ensuring a cohesive and polished appearance across diverse devices and screen sizes. One of the fundamental strengths of CSS lies in its ability to work synergistically with HTML, enabling the separation of content and design. While HTML is responsible for structuring and organizing the information on a webpage, CSS steps in to define how that content should be styled and presented. This clear distinction enhances code modularity, maintainability, and the overall efficiency of the web development process. CSS facilitates the creation of responsive designs, ensuring that web pages adapt seamlessly to different screen sizes and resolutions. This responsiveness is crucial in the era of diverse devices, ranging from desktop computers to smartphones and tablets. Through techniques like media queries and flexible grid systems, CSS empowers developers to craft layouts that gracefully adjust to the viewing environment, promoting a consistent and enjoyable user experience across platforms.

JavaScript

JavaScript is a dynamic scripting language that plays a crucial role in web development, particularly in enhancing interactivity and functionality. As a client-side language, it executes directly in the user's browser, empowering developers to create dynamic content and manipulate the Document Object Model (DOM). This enables real-time updates, responsive interfaces, and seamless handling of user input. On the front end, JavaScript enables the creation of dynamic and interactive interfaces, allowing for features like sliders, pop-ups, and live updates without requiring a page reload. It also facilitates communication with APIs (Application Programming Interfaces), enabling the retrieval and display of real-time data. In back-end development, JavaScript is commonly used with environments like Node. js, allowing developers to use the same language on both the client and server sides. This unification streamlines development workflows and promotes code reusability.

Hardware Details

The hardware requirements for the project are as follows:

1) CPU : intel Core i7 8th gen or above, AMD Ryzen 5 from 3rd gen
2) GPU : 10GB VRAM
3) RAM : 12GB

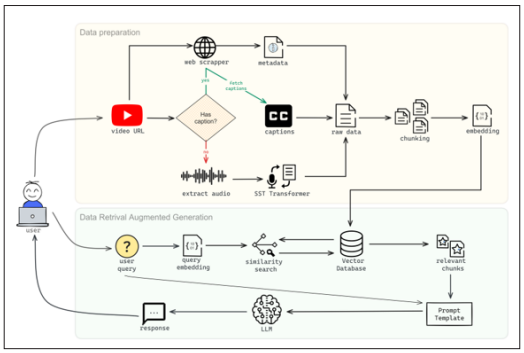

Figure 1: System Design

Methodology

Gathering Relevant Video Context

This system is designed to process video content and extract relevant information, including text and metadata. The process entails fetching video content from a designated source link and subsequently extracting textual information from the video. Additionally, metadata information is extracted from the source link through webscraping techniques. By combining these elements, both textual content and contextual details are gathered. Through the merging process, the extracted text from the video and the scraped metadata from the source link are integrated into a unified output format. This consolidated format ensures comprehensive coverage of both the content and contextual details associated with the video. Finally, the merged output is saved to a file, providing a structured and accessible representation of the combined information for future reference or analysis. This systematic approach enables efficient aggregation and organization of multimedia content and associated metadata, enhancing the accessibility and utility of the collected information.

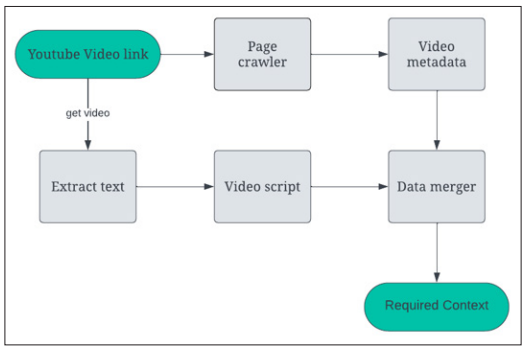

Figure 2: Data Gathering Pipeline

The transcript from YouTube Video is extracted in two ways:

Transcription via caption-scrapping

The first approach is to get the video link from a user. Then, the crawler extracts the data from the webpage. This process is complemented by scraping YouTube captions, which involves retrieving the text data embedded within the video's closed captions or subtitles.

Transcription via Speech-to-Text Models

Whisper Transformer serves as a valuable tool for extracting transcripts from videos through speech-to-text conversion. The process begins with the extraction of the audio track from the video file. It is then fed into the Whisper Transformer model for transcription. The model's architecture is based on the transformer architecture, a highly successful approach in natural language processing tasks that leverages self-attention mechanisms to capture long-range dependencies in the input data. By applying this powerful architecture to the domain of speech recognition, the Whisper Transformer model can effectively interpret the audio input and convert it into a sequence of text tokens that accurately reflect the spoken content contained within the audio track. It not only accurately transcribes the spoken words but also captures the contextual nuances and linguistic patterns present in the audio. This is achieved through the integration of large language models trained on vast amounts of text data, enabling the model to leverage grammatical rules, vocabulary, and contextual information to improve the accuracy and coherence of the transcriptions. After that, a data merger combines the video script with the metadata from the webpage. Finally, the required context is extracted from the YouTube link.

Generating Answer for User-Query

Pre-Processing the data:

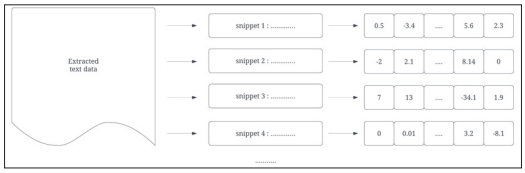

Figure 3: Pre-Processing the text data

The following is done in this stage:

1) Read the text file
2) Split the text in to small overlapping chunks
3) Encode each chunk into a vector embedding

To efficiently process large texts, it is often necessary to split them into smaller, manageable chunks. This is where the recursive text splitter comes into play. The recursive text splitter is a technique used to divide the text into smaller, overlapping chunks or segments. It recursively splits the text based on certain criteria, such as length or semantic boundaries (e.g., sentences or paragraphs). By creating overlapping chunks, the algorithm ensures that relevant context is not lost when processing the individual chunks separately. Once the text is split into chunks, each chunk needs to be encoded into a numerical representation that can be processed by machine learning models or algorithms. In this case, the Voyage-large-2 embedding scheme is used to encode the text chunks. Embeddings are dense vector representations of text, where each word, phrase, or chunk is mapped to a high- dimensional vector. These vectors capture the semantic and syntactic relationships between the text elements, allowing for efficient processing and analysis.

Data Management for Question Answering Purpose

Vector Storage

The process involves converting extracted textual information, such as text snippets, into high-dimensional vectors that capture their semantic meaning. These vectors are then efficiently stored in a specialized vector database optimized for fast retrieval and similarity search. By leveraging this approach, users can conduct efficient comparisons and retrieve related information based on semantic similarity rather than simple keyword matching. This method significantly enhances the capabilities of information retrieval systems, enabling more nuanced and contextually relevant search results. It empowers applications across various domains, including natural language processing, recommendation systems, and information retrieval, by enabling them to better understand and respond to the underlying semantic relationships within textual data.

Snippet Storage

In conjunction with storing the vectors representing extracted information, it is imperative to retain the original text snippets in a distinct database or data structure. This dual storage approach ensures access to the actual content and context associated with the vectors, enriching the retrieval process. Users can effortlessly retrieve and comprehend the extracted snippets based on their textual content, rather than relying solely on numeric vectors. By preserving the original text alongside the vectors, the system maintains the integrity of the information, facilitating a more comprehensive understanding and interpretation of the retrieved data. This approach is instrumental in various applications, including information retrieval, content analysis, and contextual understanding, as it enables users to explore and leverage the semantic meaning embedded within the textual content.

Integration and Retrieval

To facilitate seamless link age between vectors and snippets stored in different locations, a robust system can be designed to store IDs or metadata alongside the vectors, establishing a direct association between the two entities. This linkage information serves as a bridge between the numerical representations of the data and their corresponding textual content. By storing identifiers or metadata alongside the vectors, the system enables efficient retrieval of the relevant snippet based on the searched vector. When a query vector is submitted for retrieval, the system

references the associated metadata or IDs to pinpoint the location of the corresponding snippet, streamlining the retrieval process. This design ensures that users can swiftly access the textual content linked to the retrieved vectors, fostering a seamless and intuitive experience. Additionally, leveraging efficient indexing and retrieval mechanisms optimizes the performance of the system, allowing for quick and accurate retrieval of relevant snippets based on their semantic similarity to the query vector.

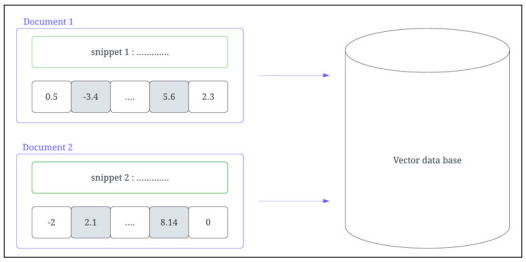

Figure 4: Data Management in Vector DB

Benefits:

- Fast and efficient retrieval of semantically similar information.
- Improved search accuracy beyond simple keyword matching.
- Scalability for handling large volumes of data.
- Flexibility for accessing both vector representations and original content.

Search information Relevant to Query:
Compute an encoding for User Query

Figure 5: Computing the Encoding of User Query

Search Vector DB for Snippets whose embedding are closest to query embedding

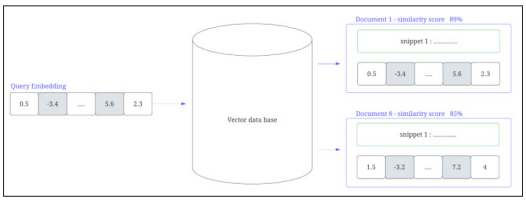

Figure 6: Searching in Vector DB

Search Vector DB for snippets whose embeddings are closest to the query embedding by utilizing the HNSW
(Hierarchical Navigable Small World) algorithm to efficiently navigate the high-dimensional vector space and identify the nearest neighbors, and then calculate the cosine similarity between the query embedding and the retrieved text chunk embeddings to obtain their respective similarity scores.

$$\cos ine\ similarity = S_C(A,B) := \text{Cos}(\theta) = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The cosine similarity score provides a measure of how similar or dissimilar the retrieved text chunks are to the query, enabling ranking or filtering of the search results based on their relevance. By combining the HNSW algorithm for approximate nearest neighbor search and the cosine similarity metric for measuring vector simi- larity, this approach allows for efficient retrieval of the most relevant text chunks from the VectorDB based on their semantic representations (embeddings) and their proximity to the query embedding in the vector space.

Answering the Question Based on Retrieved snippets

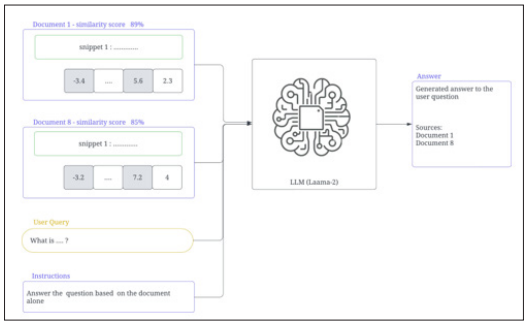

Figure 7: Answering the user query using LLM

The retrieved textual chunks, along with the original user query, are appended to the prompt template provided to the large language model (LLM). The system prompt instructs the LLM to generate a response to the user query based solely on the retrieved documentary context. The LLM then processes the provided context and generates an answer that coherently synthesizes the relevant information from the retrieved documents to address the user's query in a meaningful manner.

Result

The successful conversion from video to text signifies a notable achievement in the application of advanced natural language processing (NLP) techniques. This accomplishment, facilitated by sophisticated algorithms, demonstrates the precision with which spoken words and audio content from videos can be transcribed. However, this achievement goes beyond mere transcription; it represents a nuanced endeavor that preserves the essence of the content, capturing not only the literal meaning but also the subtleties in tone, context, and emphasis. The achieved accuracy in video-to-text conversion holds profound implications. It implies that the resulting textual representation retains the core message and key details embedded in the original video, making the information accessible in a different modality. This accessibility is crucial for a diverse audience, including individuals with hearing impairments or those who prefer consuming content through written text. Moreover, the success in video-to-text conversion provides valuable support for language translation services. The transcribed text serves as a foundation for translating the content into various languages, thereby broadening the reach of the information and enabling a more global and inclusive audience engagement. Beyond these practical applications, the successful conversion underscores the transformative potential of technology in bridging the gap between multimedia content and textual information. It lays the foundation for innovative solutions and applications where the wealth of information stored in videos can seamlessly integrate with various text- based platforms and services. This integration

enhances the discoverability and usability of video content, unlocking new possibilities for content utilization, accessibility, and user engagement.

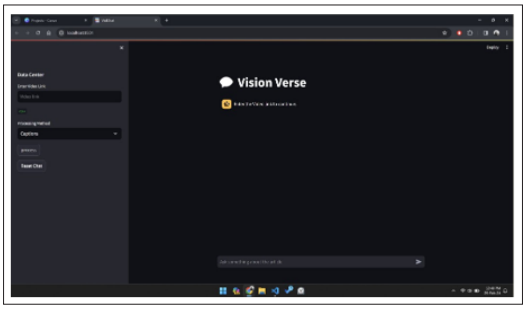

Figure 8: Frontend Of Chat Interface

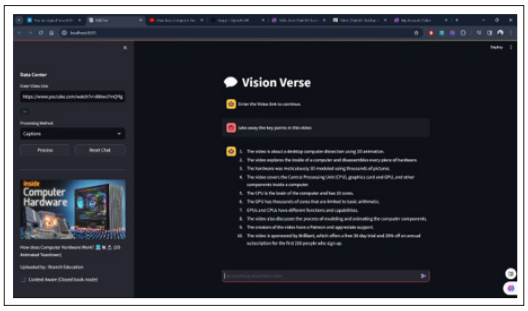

Figure 9: Answering YouTube Question : 1

When a Question is asked which is outside the scope of the provided video the system try not to give any hypothetical answer, rather it simply responds with an ”I dont know”

1) Which makes our system robust and reliable
2) Which makes the user confident in the answer provided by the system

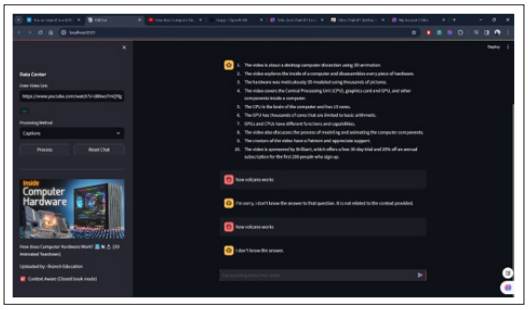

Figure 10: Response to Out of context Question with respect to video

Performance Analysis

Performance Affected by Both Retrieval and Generation

In a RAG system, the performance is dependent on the quality and relevance of the text chunks retrieved, as well as the ability of the language model (LLM) to generate a coherent and accurate response based on the retrieved information. The retrieval component, which uses a vector database, needs to efficiently retrieve the most relevant text chunks that can help answer the user's query. The generation component, performed by the LLM, must be able to process the retrieved documents and generate a response that is faithful to the context and information provided.

Quality of the Retriever

The performance of the retriever is highly dependent on the embedding scheme used to convert the text into vectors. The retriever algorithm must be able to identify the most relevant text chunks that can best answer the user's query. Effective retrieval is crucial for the overall performance of the RAG system, as the LLM can only generate a response based on the information provided by the retriever.

Importance of the Generation Part

The generation component, performed by the LLM, is the most critical part of the RAG system. The LLM must be able to process the retrieved documents and generate a response that is coherent, accurate, and relevant to the user's query. The quality of the generated response can significantly impact the overall performance of the RAG system.

Temperature Parameter in LLM

The temperature parameter in the LLM controls the level of creativity and variation in the generated text. A low temperature value (e.g., 0) makes the LLM more deterministic, leading to a unique response for a given prompt. A high temperature value (greater than 1) allows the LLM to be more creative, but this can also increase the likelihood of hallucinations (generating text that is not grounded in the input). For information retrieval applications, it is generally recommended to use a temperature value of 0 to ensure a reliable and deterministic response from the LLM.

Structuring the System Prompt

The system prompt provided to the LLM can significantly influence the quality and relevance of the generated response. The system prompt should be carefully structured to ensure that the LLM generates a response that is directly related to the context of the retrieved documents. By providing a well-designed system prompt, the LLM can be guided to produce a response that is more deterministic and aligned with the user's query.

The examples provided below highlight the importance of carefully constructing the system prompt to guide the language model's response, particularly in the context of text generation. Example video link: How does Computer Hardware Work?

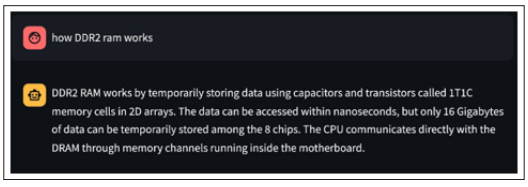

Figure 11: Hallucinated response, when the question aligns closely the context.

In the above example, the system generated a response that contained information not directly related to the specifics of how DDR2 RAM works, as the source video did not cover that topic in detail. This type of hallucinated response, where the model draws from its general knowledge rather than the provided context, can be problematic for applications that require accurate and relevant information.

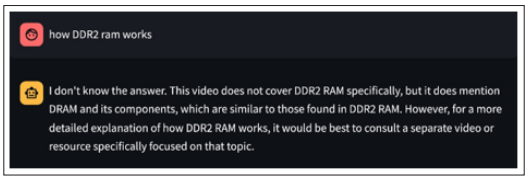

Figure 12: Highly reliable response, when the question aligns closely the context

The second example demonstrates a more appropriate system prompt and response. Here, the system acknowledges the limitations of the input material, recognizing that the video does not provide a comprehensive explanation of DDR2 RAM. Instead of attempting to generate a response beyond the scope of the available information, the system provides feedback to the user, suggesting that they consult a more specialized resource on the topic. This approach is more reliable and transparent, as it avoids making claims that cannot be substantiated by the given context.

Experimental setup

We conducted our experiments on a RAG system trained on a large corpus of question-answer pairs and augmented with a retrieval mechanism to access relevant information from YouTube videos. The system was evaluated on a dataset of 100 out-of-context queries, where the queries were deliberately designed to be challenging and require external knowledge for accurate answering.

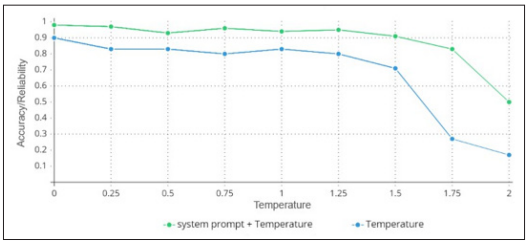

Figure 13: Rejection of out of context query

We evaluated the system's performance at different temperature values ranging from 0 to 2, with increments of 0.25. For each temperature value, we generated 100 samples and calculated the accuracy and reliability metrics based on human evaluations. Additionally, we explored the impact of varying both the system prompt and temperature simultaneously to assess their combined effect on the system's performance. Our experimental results are presented in the form of a plot, which clearly illustrates the relationship between temperature and the system's accuracy/reliability in handling out-of-context queries. As expected, the "Temperature" line indicates a steady decline in the system's accuracy/reliability as the temperature increases.

This behavior is consistent with the understanding that higher temperatures introduce more randomness and diversity in the generated outputs, potentially compromising the coherence and accuracy of the responses. These findings highlight the importance of careful temperature selection and prompt engineering in RAG systems, as higher temperatures can introduce more diverse but potentially less accurate and coherent responses. Overall, the performance of a RAG system for answering YouTube video queries based on extracted text depends on the quality of the retriever, the effectiveness of the LLM in generating relevant responses, and the careful configuration of the system parameters, such as the temperature and the system prompt.

Future Scope

In the realm of personalized learning, our project is positioned for significant expansion and refinement. Leveraging machine learning algorithms, we envision a future where video summaries and chatbot interactions are tailored to individual user preferences and learning styles, enhancing engagement and comprehension. This personalized approach promises more effective educational experiences. Additionally, we prioritize inclusivity and accessibility. Enabling users to access video content in their native languages fosters a more inter- connected and inclusive learning community. Integration with Learning Management Systems (LMS) streamlines content delivery and enhances student engagement within established learning environments [8-13].

Enhancing usability is a key focus. Voice-based interaction capabilities empower users to interact with the system naturally, enhancing accessibility and user experience. Robust analytics features provide insights into engagement patterns, preferences, and knowledge retention, enabling continuous improvement. Fostering collaboration is essential. Introducing collaboration tools within the platform facilitates group discussions, collaborative learning activities, and knowledge sharing, creating a dynamic learning community. Exploring Augmented Reality (AR) and Virtual Reality (VR) integration offers immersive learning experiences, enriching engagement with video content and fostering experiential learning and knowledge acquisition.

Conclusion

In today's digital era, video content stands out as a prominent medium for sharing information, entertaining audiences, and facilitating education. However, the sheer length of video content often poses challenges for users in terms of comprehension and time constraints. To address these challenges, our project introduces a forward-thinking solution that leverages state-of-the-art language models to revolutionize the way users interact with video content. At the heart of our innovation lies the ability to transform extensive video content into concise and digestible text summaries. By tapping into the capabilities of LLM, we ensure that the core message and key details of the video are accurately captured and retained in the summarized text. This process significantly enhances the efficiency of content consumption by providing users with a quick, readable overview of the video contents, saving them valuable time and effort. Furthermore, our solution goes beyond mere summarization by introducing an interactive chatbot companion. Powered by natural language understanding and generation, the chatbot enables users to engage with the video content in a dynamic and meaningful way. Users can ask questions, seek clarifications, or delve deeper into specific aspects of the video, fostering better understanding and encouraging active participation. By combining the benefits of interactive chatbot and video summarization technologies, our solution offers users a dynamic and engaging means to interact with video content. Whether in educational settings, research environments, our innovation holds significant potential for streamlining information dissemination and enhancing user experiences. In essence, our project represents a pivotal

advancement in the realm of digital content consumption, paving the way for more efficient and accessible interactions with video content in the digital age.

Declarations

Availability of data and materials

The code of the program is available on GitHub https://github.com/abhiram-ar/VidChat.

Competing interests

Not applicable

Funding

Not applicable

References

1. Saraswathi M, Ronit VV, Pranav SS. Implementation of Video and Audio to Text Converter International Journal of Research Publication and Reviews. 2023. 4: 1204-1208.
2. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, et al. Language Models are Few- Shot Learners. Advances in neural information processing systems. 2020. 33: 1877-901.
3. Zhou X. Deep Learning for Semantic Video Indexing: A Review. ACM Computing Surveys (CSUR). 2018. 51: 1-36.
4. https://www.youtube.com/watch?v=TQQlZhbC5ps
5. Qian H. User Engagement Analysis in Online Video Platform: A Literature Review. Journal of Information Science Theory and Practice. 2019. 7: 60-73.
6. Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020.
7. Yu Malkov A,Yashunin DA. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. 2016.
8. Radford A. Robust Speech Recognition via Large-Scale Weak Supervision. 2022.
9. Zhongqiang Huang, Mary Harper. Empirical Methods in Natural Language Processing. 2009.
10. David McClosky, Eugene Charniak, Mark Johnson. Effective self-training for parsing. In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference. 2006. 152-159.
11. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, et al. Attention is all you need. 2023.
12. Jon Saad-Falcon, Joe Barrow, Alexa Siu , Ani Nenkova, David Seunghyun Yoon, et al. PDFTriage: Question Answering over Long, Structured Documents. 2023.
13. Adnan Arefeen MD, Biplob Debnath, Srimat Chakradhar. Lean Context: Cost-Efficient Domain Specific Question Answering Using LLMs. 2023.